Large Scale Authorship Attribution of Online Reviews

Prasha Shrestha, Arjun Mukherjee, and Thamar Solorio

Department of Computer Science
University of Houston
Houston, TX, 77004
pshrestha3@uh.edu {arjun, solorio}@cs.uh.edu

Abstract. Traditional authorship attribution methods focus on the scenario of a limited number of authors writing long pieces of text. These methods are engineered to work on a small number of authors and generally do not scale well to a corpus of online reviews where the candidate set of authors is large. However, attribution of online reviews is important as they are replete with deception and spam. We evaluate a new large scale approach for predicting authorship via the task of verification on online reviews. Our evaluation considers a large number of possible candidate authors seen to date. Our results show that multiple verification models can be successfully combined to associate reviews with their correct author in more than 78% of the time. We propose that our approach can be used to slow down or deter the number of deceptive reviews in the wild.

1 Introduction

With almost everything being online, there has been what can only be called a deluge of social media data. For example, Amazon has 244 million active users [1] and Yelp had 83 million unique visitors per month in the fourth quarter of 2015 [2]. Much of the previous research on traditional authorship attribution deals with a small set (≤ 10) of authors [3, 4]. Some more recent researchers have worked on a relatively larger number of authors [5, 6]. But in order to keep up with the increase in online data, we need scalable approaches that can work for companies like Amazon and Yelp.

Given a set of authors, authorship attribution(AA) is the task of figuring out who, if any of them is the actual author of a piece of text. AA is not a new field as it has been around from the start of 19th century [7]. But AA on social media data is fairly new. Among various forms of social media data, it is specially important to focus on AA for the domain of product reviews because it contains a lot of fake reviews and spam [8, 9]. A single user might create multiple accounts in order to write outstanding reviews for a product in order to promote it. In the same way, negative product reviews could be written for the sake of hindering a competitor's product [10]. Such users are likely to have multiple accounts, a legitimate account and one or more fake accounts. AA can help to detect if two accounts belong to the same author or not and with some modifications verification can also help detect when more than one author is writing reviews under the same user id. AA methods can also be extended to include background author detection to predict not only if a text has been written by an author but also if the text is written by none of the authors.

Most traditional authorship attribution tasks are performed on long texts such as books. There has been a growing interest on authorship attribution of social media data, but most of it has been focused on blog data [5]. Reviews are different from blogs in that reviews are generally shorter. Also, the topic of the reviews will be different from product to product whereas a blogger tends to be focused on a fixed set of topics. It is not clear that even internet scale attribution [11] can be adapted to online reviews. Authorship attribution on reviews is more challenging as:

- the number of candidate authors is very high (ten to hundreds of magnitude larger than most existing work)
- reviews are usually very short as compared to books or blogs
- even the reviews written by the same author differ in topic because users typically write one review per product purchased
- while spamming, authors deliberately try to alter their writing to avoid getting caught

Our work addresses AA on larger author sets and on noisy/short review texts. In this paper, we use two new datasets of online reviews that can be used to develop and benchmark new approaches for authorship attribution at a large scale. One of the datasets consists of product reviews from Amazon and the other one contains reviews from Yelp. We first present a verification technique through which we will perform the attribution of reviews. Our contributions are: first we present a large review dataset that can be used to benchmark author verification, attribution and background (out-of-set) author detection. We also present our approach to the problem of verification and attribution in datasets having a large author set.

2 Related Work

Our work follows that in Koppel and Winter (2014), where they reason that any AA problem can be broken into a set of Author Verification(AV) problems [5]. Conversely, they also show that an AV problem can also be converted in to a many-candidates problem. In order to obtain these candidates, their AV system generates impostors from documents of the same genre as a given document. If two documents are consistently more similar to each other than to the impostors across different ngram feature sets, then they are likely to be written by the same author. By using this method, they obtain more than 90% optimal accuracy for 500 pairs of long, 2000-word documents.

Qian et al. (2015) also perform AA on online reviews. They first generate document based features and convert them to various text similarity features between two reviews and train on these features. Their document based features consist of various frequency based features, writing density features and vocabulary richness features. They then define their own formulas that use these document feature values of two reviews in order to produce features representing similarity between the two reviews. Their AA method computes these similarity features for a test review and individual reviews from all candidate authors. They train their model on these similarity features to predict if the two reviews are written by a single author or by different authors. They obtain scores from this model for various reviews of an author and combine these scores to obtain the

actual author. Their best method obtained in a range of around 40% to 83% accuracy for a range of 2 to 100 candidate authors.

Seroussi et al. (2014) used topic modeling to generate author representation. Their main idea is based on distinguishing between document/topic specific words and author specific words. They experimented with five different datasets covering a wide variety of topics, number of authors and amount of text per author. Along with count judgements, blog posts and emails, there are also two datasets on Internet Movie database (IMDb) movie reviews. They compared their system with a baseline of token frequency counts used as features to train a SVM model. Their proposed system outperformed the baseline in four out of the five datasets. The baseline beats their system by a close margin on a movie review dataset consisting of reviews from prolific IMDb users. However, their system beats the baseline on the reviews dataset consisting of reviews from more than 22,000 random IMDb users.

Stamatatos (2009) observe that AA researchers have used various lexical, syntactic and semantic features to capture the style of an author [7]. Among these features, lexical features are the most prominently employed features in authorship attribution systems and especially character n-gram based features have given good performance. Character n-grams are also among our features. They also distinguish between two types of attribution methods: profile based approach where all instances of an author's writings are combined to create a single profile for an author and instance based approach where each instance of an author's writings are treated separately. We use the instance based method and choose to combine the results from these instances instead.

Eder (2015) try to analyze how the size of an author's text relates to the performance of the task of authorship attribution by using the Delta method [12]. They perform their experiments on separate datasets of English, German, Polish, Latin, Greek and Hungarian language novels. They find that for all languages, the performance on the AA task generally improves with larger amount of text. But after certain length, which the author found to be around 5,000 words, the performance starts to saturate and might even decrease a little. Since we are dealing with reviews, it is very rare for our data to reach this number. The average number of words per review in our dataset is only 233.46. They also analyze the difference in performance when using consecutive blocks of text versus randomly chosen bag of words. Interestingly, they found that using the randomly chosen bag of words gives better performance. They also tried ngrams with variable numbers of n. They found that the performance steadily decreases with increasing n. This finding aligns with previous researches [7] and we also limit our n to less than 4.

3 Review Datasets

We have two review datasets for performing AA on a large number of authors and/or on online data. The Amazon dataset contains the reviews written by the authors for different products on Amazon while the Yelp dataset comprises restaurant and hotel reviews. Along with the text of the reviews, there are other attributes of the reviews in both datasets that might be useful for AA. The Amazon reviews contain information about the reviewer id, date posted, star rating and helpful count of the reviews. The Yelp reviews also contain reviewer id, star rating and date posted, along with useful, funny

and cool counts given to the reviews. The Amazon reviews were posted between June 1996 to October 2012 and the Yelp reviews between January 2006 to September 2012.

Table 1: Number of authors with $\geq x$ reviews

Criteria ($\geq x$)	Amazon	Yelp Hotels	Yelp Restaurants
50	8,171	2,450	3,174
25	15,772	3,064	5,322
1	123,967	5,132	35,392

As shown in Table 1, the datasets have a very large number of authors with the Amazon dataset having the highest number of authors. Apart from having a large number of candidate authors, on average there are only around 240 tokens per review for Amazon dataset and 106 tokens per review for the Yelp dataset. Some of the authors only have a few reviews to their name. It is very hard to perform verification and attribution when there is insufficient text. As can be seen in Table 1, only a fraction of the authors have 50 reviews or more, although the number is still high. Our final datasets only consist of the prolific authors, who have written at least 50 reviews each. This also helps us to winnow out most of the sockpuppet accounts since usually sockpuppet accounts only contain a small number of reviews [10]. As such, we use reviewer and author interchangeably. [1]

4 Attribution via Verification

When the number of authors is very high, as in our case, it is unrealistic to try to train a single, combined, multi-class AA model for all the authors. It is much more manageable to break down the problem into smaller pieces. Thus, we approach the problem of AA on a large set of authors by training individual verification models or verifiers for each of the authors. We begin with a set of n authors $A = \{a_1, ..., a_n\}$ and their set of documents (reviews) $D = \{D_1, ..., D_n\}$. Here D_i is a set of reviews written by a_i. We extract features from these reviews and then train n separate verifiers in which each of the reviews acts as an instance of an author's writing. An author verifier performs a single task of predicting if a given review is in fact written by the same author or not.

In order to train these verifiers we perform review data selection in the following way: For each of the authors we define their own set of positive and negative reviews. Here we use positive and negative in the sense that all the reviews belonging to an author are positive reviews for him/her and reviews written by all other authors in our dataset comprise the negative reviews for that author. If we are training a verification model for author a_i, then all of a_i's reviews are his/her positive reviews. All the reviews from the other $n - 1$ authors are possible candidates for negative reviews. For each verifier, we create a balanced training and test dataset. We will discuss later how this is not a

[1] The datasets can be obtained at http://ritual.uh.edu/resources/

limitation for our approach. Since there are a large number of candidates for negative reviews, we need to perform negative review selection.

Algorithm 1: Training and Test Set Selection

Assume A is a set of authors and $|A|= n$
For each author $a_i \in A$:
$S^+_{a_i} = D_i$
that is, all documents from author a are positive instances in the verification case for author a.
Split $S^+_{a_i} = \{S^+_{a_i\_train}, S^+_{a_i\_test}\} = \{80 : 10\}$ from $S^+_{a_i}$

Generate negative samples as follows:
Negative Open Set(NOS):
$S^-_{a_i\_train} \subset \{s|s \in D_j, j \neq i\}, j = 1, ..., n/2;$
$S^-_{a_i\_test} \subset \{s|s \in D_j, j \neq i\}, j = n/2 + 1, ..., n$

Negative Random Set(NRS):
$S^-_{a_i} = D - D_i$
$S^-_{a_i\_train} \subset S^-_{ia}$; $S^-_{a_i\_test} \subset S^-_{a_i}$
subject to the constraints:
$|S^-_{a_i\_train}|= |S^+_{a_i\_train}|$
$|S^-_{a_i\_test}|= |S^+_{a_i\_test}|$
$S^-_{a_i\_train} \bigcap S^-_{a_i\_test} = \phi$

The training and test set selection is shown in Algorithm 1 and described here. Selection of the positive set of documents is straightforward. We take 80% of all of the reviews of an author as the positive training set and 10% as the positive test set. We hold out 10% for an analysis that we will explain in Section 5.1. For the selection of negative set of reviews, we tried two different methods. Our motivation behind this is to find out whether having reviews in the training set from those authors whose reviews are also present in the test set makes a difference or not. We describe the two different methods for the negative set selection below.

4.1 Negative Open Set (NOS)

In the NOS method, there is no overlap between the authors whose reviews appear in the training set and the authors whose reviews are present in the test set. We first divide the $n - 1$ authors into two sets of $\lfloor (n - 1)/2 \rfloor$ authors each. Then we select $|S^+_{a_i\_train}|$ reviews from among the reviews written by the authors in the first set as the negative training reviews and $|S^+_{a_i\_test}|$ reviews from those written by the authors in the second set as the negative test reviews.

4.2 Negative Random Set (NRS)

In the NRS method, there might be an overlap between the authors whose reviews are present in the training set and the authors whose reviews are in the test set. Here, both

the negative training and test reviews can come from all $n-1$ authors. We will select $|S^{+}_{a_i\_train}|$ reviews for training and $|S^{+}_{a_i\_test}|$ for test set from the pool of all $n-1$ authors' reviews.

4.3 Features

We use the features that have already been tested in previous AA research or in related fields such as intrinsic plagiarism detection [13] and anomaly detection [14]. We also performed a simple preprocessing step of removing any URLs and converting the text to lowercase before extracting the features. We did not use other preprocessing steps such as stopword removal because stopwords can be important in AA and they are a part of some of our features as we will describe below. Our features include the following:

Lexical: These consist of word unigrams and character unigrams, bigrams, and trigrams. An author is likely to show preferences for certain words more than other authors, which can help distinguish him/her. Similarly, character ngrams capture the writing style of an author and have been used by previous researches successfully for authorship attribution [5, 4, 7].

Syntactic: We extract part of speech (POS) tags as well as chunks by using the tagger and chunker available in Apache OpenNLP. We try to model the style of the author by using POS and chunk unigrams, bigrams, and trigrams.

Writing Density: Authors can also be distinguished by their writing density. For our experiments, these include the average number of characters per word, the average number of syllables per word, and the average number of words per sentence.

Readability: The complexity of a piece of text varies from author to author. We use standard readability indices to measure this. We use FleschKincaid grade level [15, 16], Gunning fog index [17], Yule's K measure, and Honore R measure [13] as our features.

Part of Speech (POS) Trigram Diversity: This is yet another feature that captures the style of an author [14]. It is quite simply the number of unique POS trigrams normalized by the total number of POS trigrams in the text.

Stopword Frequency: This is another stylistic feature and it measures the proportion of stopwords in an author's text. It is measured as the total number of stopwords in a piece of text divided by the overall count of the words in the text.

Average Word Frequency Class: This is a measure of how likely an author is to use unique words that are not frequently used in a language [18]. A word is assigned frequency class according to how likely it is to appear in a corpus. Meyer zu Eissen and Stein (2004) used the Sydney Morning Herald Corpus to obtain the word frequency class for about one hundred thousand words. They range from 0 for 'the', the most common word in the corpus to 19 for very uncommon words. We took the frequency class for every word of an author's text present in these one hundred thousand words and then normalized it by the total number of words in the text.

Table 2: Author verification results showing macro-averaged values

Dataset	Method	Positive Class			Negative Class			Accuracy
		Precision	Recall	F-score	Precision	Recall	F-score	
Amazon Reviews	NOS	0.8674	0.9165	0.8846	0.9193	0.8423	0.8696	87.94
Amazon Reviews	NRS	0.8600	0.9162	0.8806	0.9187	0.8331	0.8639	87.47
Yelp Hotel	NOS	0.8517	0.8921	0.8678	0.8915	0.8358	0.8579	86.39
Yelp Hotel	NRS	0.8636	0.8916	0.8732	0.8927	0.8495	0.8656	87.05
Yelp Restaurant	NOS	0.8595	0.8757	0.8617	0.8804	0.8449	0.8557	86.03
Yelp Restaurant	NRS	0.8567	0.8799	0.8628	0.8825	0.8401	0.854	86.00

4.4 Authorship Attribution

To perform AA on a review, we first pass it to all our author verifiers. Each of the author verifiers gives us a value (P_i) representing the probability of a review being written by that author. We use logistic regression as the classifier for all of our verifiers. The probability from our verifiers is simply the probability of a review belonging to an author's class as given by logistic regression. For a review document r, we will have a set $P = \{P_1, ...P_n\}$ of such probabilities, where $P_i = P(a_i|r)$ i.e. the probability that the author of review r is author a_i. With these probabilities, we perform the final attribution in two ways.

Attribution per review: We simply choose the author whose model gives the highest probability value for a given review as the author for that review. In other words, if $max(P) = P_i$ for review r, then a_i is the author of review r.

Collective attribution per unknown author: The second method combines results from the first method. All online reviews have reviewer ids associated with them. We perform AA on the reviews having same the reviewer id (R) collectively. The intuition behind this method is that the author who is most frequently predicted by the attribution per review method as the actual author for these reviews is likely the actual author for all of these reviews. This is a valid formulation because reviewer ids are not related to author identities and doing this is similar to aggregating all the reviews of an author and performing AA on the aggregated text. We take the predictions from the attribution per review method for all the reviews having the same reviewer id R and use them as votes. The author obtaining the highest number of votes is the author of all these reviews as shown in Equation 1.

$$voting\_pred(R) = \arg\max_{a_i}(count(attribution\_per\_review(r_j) = a_i)), \quad \text{where } r_j \epsilon R \tag{1}$$

5 Results and Analysis

We performed three separate experiments on 1,000 authors from the Amazon dataset and 500 authors each from the Yelp hotel and restaurant reviews datasets. An author

verifier has separate precision, recall, f-score and accuracy values. Table 2 shows the macro-averaged values for all these four metrics across the author verifiers. As evidenced by the results, our system performs fairly well in this task. We achieve similar f-scores for both the positive and negative classes. But the precision and recall values are different. The precision for the negative class is higher while the recall is higher for the positive class. This might happen because all of the positive training examples belong to the same author but the negative reviews belong to different authors.

The NOS method gives us slightly better results, although the difference is negligible. This shows that even if we use the reviews from an author in both negative training and test, it does not affect the final results. Since we obtain good results in this task, it makes sense for us to use these results in order to perform AA.

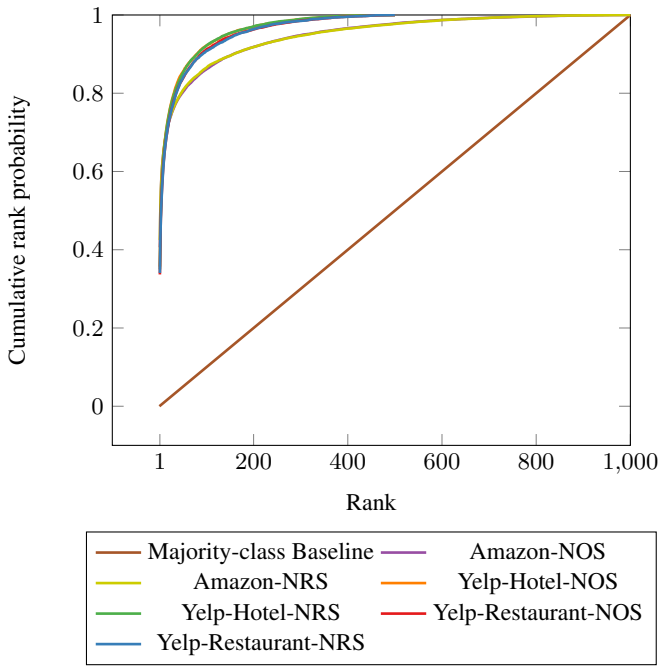


Fig. 1: Cumulative rank curve for the ranks of the actual authors

5.1 Performance on the Cumulative Rank Curve

Before AA, we perform a separate experiment to assess how good our author verifiers actually are. We do this on the 10% of the positive reviews that we held out. For each review r, we obtain the probabilities as mentioned in Section 4.4 from all author verifiers ($\{P_1, ...P_n\}$) and then rank these probabilities from the highest to the lowest. If

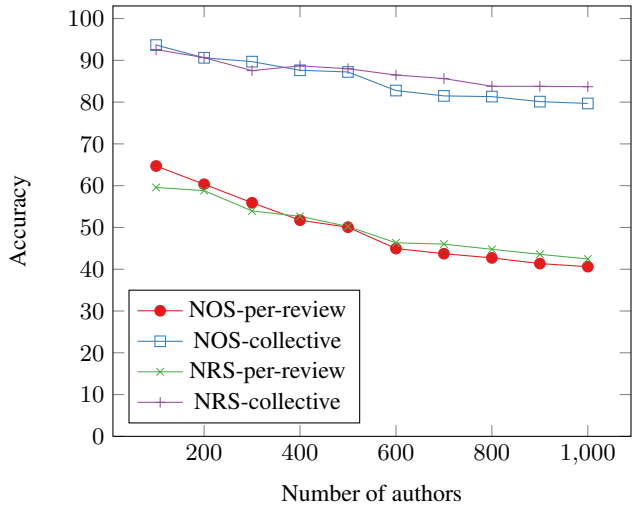
the author of r is a_i, we then get the rank obtained by the probability P_i. The rank value will be in the range from $1...n$. After we obtain this rank value for all of the reviews, we get a count of the number of reviews that fall under each rank. From these counts we finally calculate the cumulative probabilities of getting these ranks. The cumulative probability for rank k is the probability of an author verifier of author a_i obtaining rank $\leq k$ for reviews written by a_i.

This probability represents how likely our author verifiers are to get the highest probability values for reviews written by their author. For example, in an ideal scenario, all author verifiers would produce the highest probability for reviews written by their author and the probability for rank 1 would be 1. We create a plot of these probabilities against the ranks. The results from this experiment is very interesting as shown in Figure 1. The cumulative probability for rank 1 is 0.4245 for the Amazon dataset. This means that for more than 40% of the reviews, the author verifier of the actual author of the reviews obtained the highest probability for those reviews among all 1000 verifiers. For the Yelp hotel and restaurant datasets, the cumulative probability for rank 1 is 0.3568 and 0.3406 respectively. The number steadily increases such that when we get to rank 50, the cumulative probability for all datasets is higher than 0.80. This means that for more than 80% of the reviews, the verifier for the actual author is ranked 50 or higher. This provides further motivation for performing AA by using these verifiers. Another observation from this experiment is that again, NOS and NRS have similar results with overlapping curves.

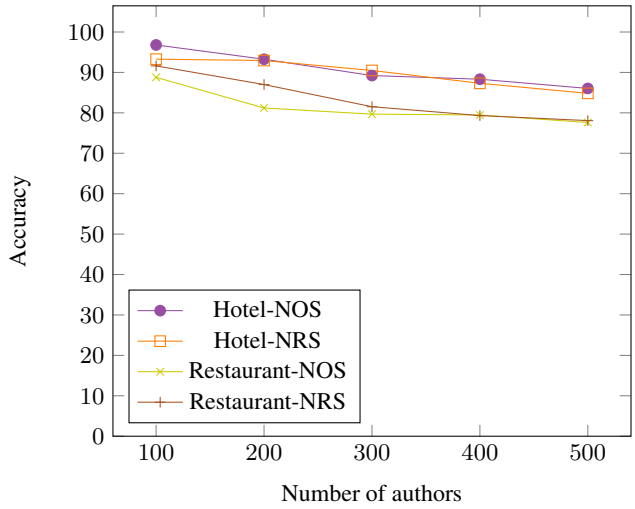
5.2 Authorship Attribution

The results for the attribution task are shown in Figure 2. As mentioned before, the trained models for all of the experiments remain the same. We performed this test on different number of authors for up to a 1000 and 500 authors for Amazon and Yelp datasets respectively. For the attribution per review method, there were ties in the highest probability values for a very small percentage (close to 0.1%) of the reviews. In these cases, if the actual author was among the authors tied for the first place, we counted that as a correct prediction. Otherwise, we selected an author from the tied authors randomly as the predicted author.

As seen for the Amazon dataset (Figure 2a), the results from the attribution per review method is not very satisfactory and they deteriorate quickly when we increase the number of authors. This is understandable as with growing contentions of candidate authors, the probability space in this method becomes sparse. But for the collective attribution method, all of the results are either close to or above 80%. The method scales well and is fairly stable even when the number of authors is increased. Like the attribution per review method, there were ties in this case too, again for a small percentage (close to 0.8%) of the authors. We used the same method as before to resolve the ties. Again, the NOS and NRS methods give similar performance even for this AA task, showing that having an already seen author in the test set does not make much difference. We also found that the accuracy is positively correlated with the total number of reviews of an author. The dataset for this experiment is very imbalanced, with less than 0.5% of the documents belonging to the positive class for an author and we still



(a) Results from the attribution per review and the collective attribution per unknown author method for Amazon reviews



(b) Results from the collective attribution per unknown author method for Yelp reviews

Fig. 2: F1-scores for the AA task for NOS and NRS

perform well. This resembles a real world scenario where there will be a lot less text written by an author as compared to all of the text written by everybody else.

Since the collective attribution method worked so well for the Amazon dataset, we tried the same method on the Yelp datasets as well. The results can be seen in Figure 2b. For 100 authors in the Yelp Hotel dataset, we obtain 96.79% accuracy in the NOS setting, which is very high considering the number of authors. Other observations

are similar to what we found for the Amazon dataset. The method scales well and the results stay either above or near the 80% mark. Again, the NOS and NRS methods perform similarly here as well. Although both datasets are reviews, they have differences between them because the review subjects are very different. But our method performs consistently well for Amazon product reviews as well as for Yelp restaurant and hotel reviews. This also shows that our method works well across different topics.

6 Conclusion and Future Work

We have presented a method to perform authorship attribution on reviews in a large scale. We proposed to use individual author verifiers to solve the AA problem on a large number of authors. We were able to build a scalable approach especially geared towards large datasets. Our method was able to obtain good accuracy even when the number of authors is large and the loss in accuracy as we go from 100 authors to 1000 authors was not very high. The first reason why our AA method worked well is that our author verifiers themselves perform very well. They can very well be used separately for authorship verification. The second reason why our approach worked is the collective attribution method we applied to combine the results from our individual verifiers. Our final system gives a good performance and is very well suited for attribution of online reviews. Not only that, the large review datasets that we have used in this paper can be useful as benchmark datasets for future large scale authorship attribution research. In our ongoing work, we are looking at extending our method in order to perform background author detection.

Acknowledgments

This research has been partially supported by NSF award No. 1462141.

References

1. Amazon Media Group: About Amazon Media Group. `http://www.amazon.com/b?ie=UTF8&node=8445211011` (2015) [Online; accessed Feb 07, 2016].
2. The Yelp Blog: About. `http://www.yelp.com/about` (2015) [Online; accessed Feb 07, 2016].
3. Kešelj, V., Peng, F., Cercone, N., Thomas, C.: N-gram-based author profiles for authorship attribution. In: Proceedings of the Conference of Pacific Association for Computational Linguistics, PACLING. Volume 3. (2003) 255–264
4. Koppel, M., Schler, J.: Authorship verification as a one-class classification problem. In: Proceedings of the Twenty-first International Conference on Machine Learning. ICML '04, New York, NY, USA, ACM (2004) 62–
5. Koppel, M., Winter, Y.: Determining if two documents are written by the same author. Journal of the Association for Information Science and Technology **65** (2014) 178–187
6. Qian, T.Y., Liu, B., Li, Q., Si, J.: Review authorship attribution in a similarity space. Journal of Computer Science and Technology **30** (2015) 200–213
7. Stamatatos, E.: A survey of modern authorship attribution methods. Journal of the American Society for Information Science and Technology **60** (2009) 538–556

8. Lappas, T.: Fake reviews: The malicious perspective. In Bouma, G., Ittoo, A., Mtais, E., Wortmann, H., eds.: Natural Language Processing and Information Systems. Volume 7337 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2012) 23–34
9. Jindal, N., Liu, B.: Opinion spam and analysis. In: Proceedings of the 2008 International Conference on Web Search and Data Mining. WSDM '08, New York, NY, USA, ACM (2008) 219–230
10. Mukherjee, A., Liu, B., Glance, N.: Spotting fake reviewer groups in consumer reviews. In: Proceedings of the 21st International Conference on World Wide Web. WWW '12, New York, NY, USA, ACM (2012) 191–200
11. Narayanan, A., Paskov, H., Gong, N., Bethencourt, J., Stefanov, E., Shin, E., Song, D.: On the feasibility of internet-scale author identification. In: Security and Privacy (SP), 2012 IEEE Symposium on. (2012) 300–314
12. Burrows, J.: delta: a measure of stylistic difference and a guide to likely authorship. Literary and Linguistic Computing **17** (2002) 267–287
13. Stein, B., Lipka, N., Prettenhofer, P.: Intrinsic plagiarism analysis. Language Resources and Evaluation **45** (2011) 63–82
14. Guthrie, D., Guthrie, L., Allison, B., Wilks, Y.: Unsupervised anomaly detection. In: Proceedings of the 20th International Joint Conference on Artifical Intelligence. IJCAI'07, San Francisco, CA, USA, Morgan Kaufmann Publishers Inc. (2007) 1624–1628
15. Flesch, R.: A new readability yardstick. Journal of Applied Psychology **32** (1948) 221–223
16. Kincaid, J.P., Fishburne Jr, R.P., Rogers, R.L., Chissom, B.S.: Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel. Technical report (1975)
17. Gunning, R.: The technique of clear writing. (1952)
18. Meyer zu Eissen, S., Stein, B.: Genre classification of web pages. In Biundo, S., Frhwirth, T., Palm, G., eds.: KI 2004: Advances in Artificial Intelligence. Volume 3238 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2004) 256–269